Style Change Detection

Based On Writing Style Similarity

-- Zhijie Zhang

From China

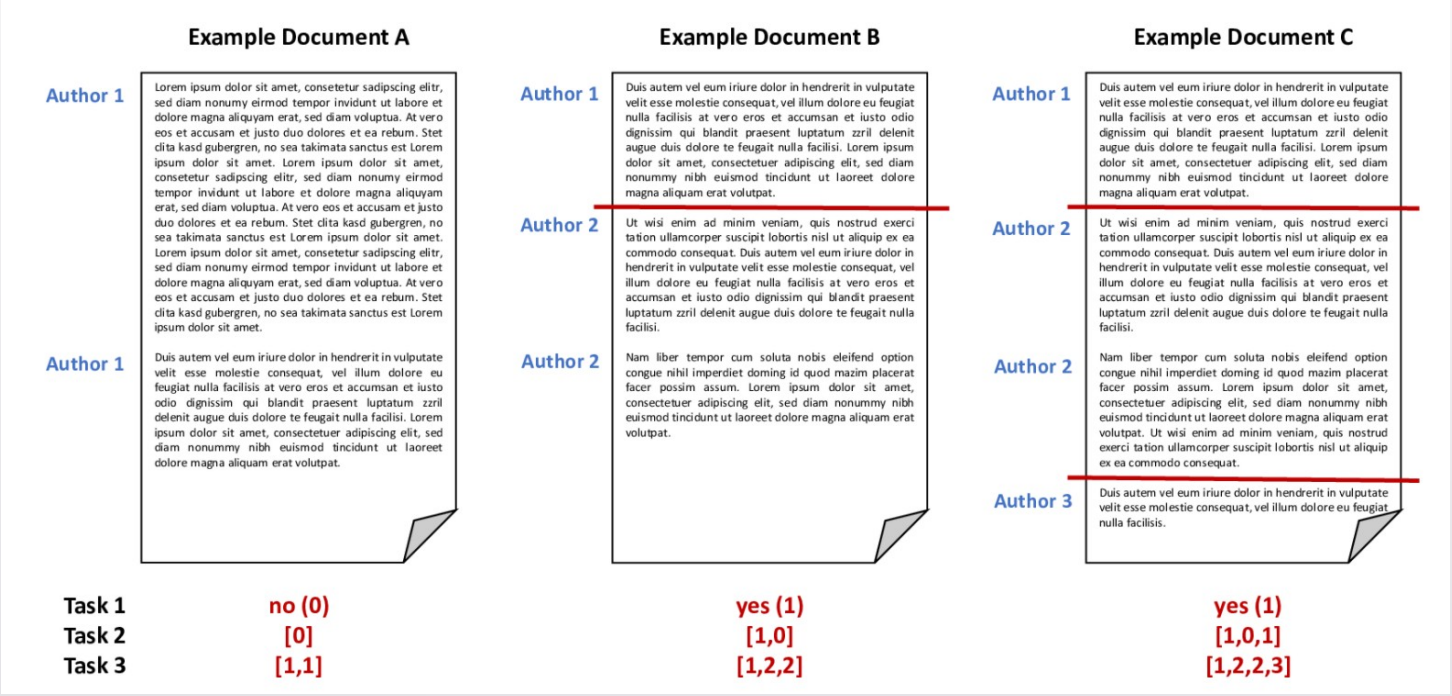
Style Change Detection

- **Task1: Detect if the document has multiple authors**
- **Task2: Find out where the style changes have occurred**
- **Task3: Label the author identifier for each paragraph of the document**

Our method :

Let three tasks can be achieved under a uniform framework by utilizing the paragraphs writing style similarity

Wrting Style Similarity



What we use is the paragraph writing style similarity. We need to estimate the similarity between two paragraphs.

The similarity label will be 0 if the writing style between two paragraphs has not changed, or it will be 1.

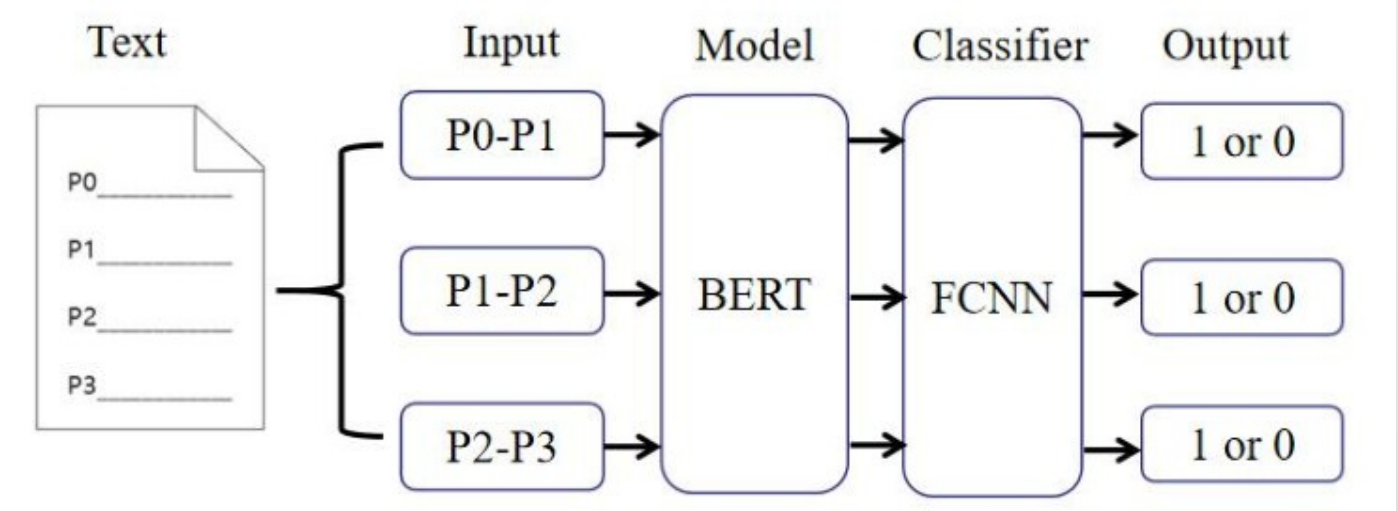
Pretrained Model BERT

- **So, how do we estimate the writing style similarity?**

We adopt the popular pretrained model BERT to extract two paragraphs features.

And then use FCNN (Fully Connected Neural Network) Classifier to perform binary classification.

Finally, we can output the similarity labels



Task2 label to Task1 label

After we got the similarity labels, we can use them skillfully in the three tasks.

- **Task1: Detect if the document has multiple authors**
- **Task2: Find out where the style changes have occurred**

After analyzing the task definitions, we found that the text will at least be two authors if the corresponding task2 label includes 1, and the corresponding task1 label will be 1. Otherwise, the task1 label will be 0.

e.g.

Task2 label: [1,0,1] [0,0,0] [1,1,0]

↓infer

Task1 label: 1 0 1

Task3 label to Task3-binary label

- **Task3: Label the author identifier for each paragraph of the document**

In task3, the paragraphs-author label includes 1,2,3,4.
e.g. [1,1] [1,2,3] [1,2,3,2,4]

These kind of labels are not friendly to our similarity labels.

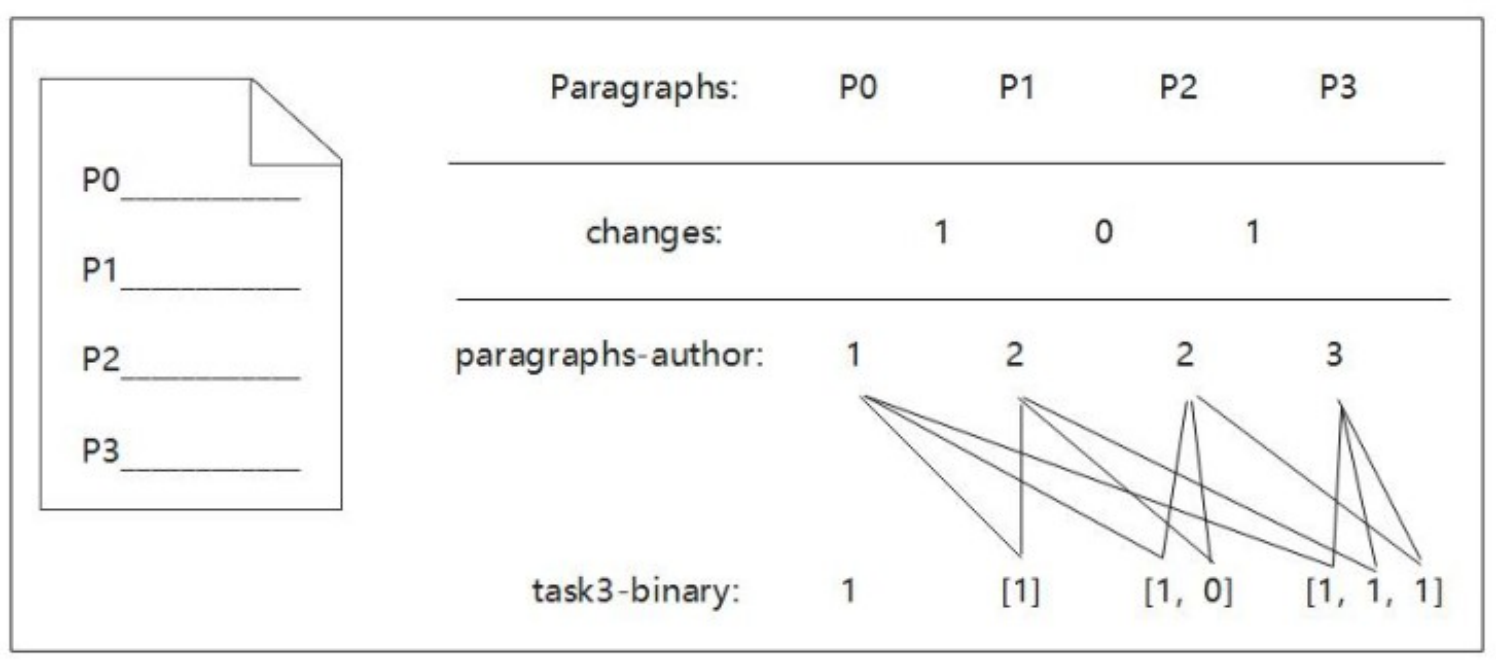
In order to let three tasks can be achieved under a uniform framework, we convert the task3 label to the binary label, which is called the task3-binary label.

SO... How do we convert the task3 labels to task3-binary labels?

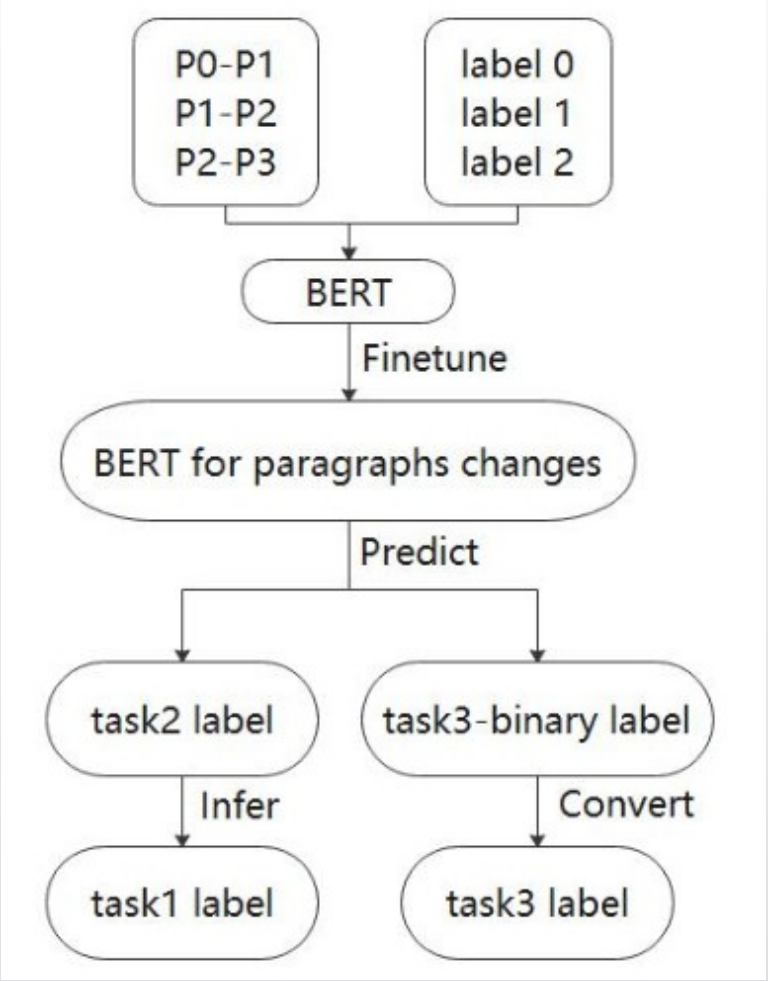
Task3 label to Task3-binary label

- **In terms of task3-binary label, the principle of the converting is shown in the Figure**

In a document, four paragraphs are denoted as P0, P1, P2, and P3 separately.
Then judge whether, for each paragraph and each of its preceding paragraphs, a style change occurs.
In the task3-binary label, the label of P0 is always 1.



Procedure


P0-P1
P1-P2
P2-P3
label 0
label 1
label 2
BERT
Finetune
BERT for paragraphs changes
Predict
task2 label
task3-binary label
Infer
Convert
task1 label
task3 label

Finetune

As we know, the pretrained model can use the training dataset and validation dataset to finetune.

Our finetune setting:

- **training dataset: task3-binary labels and their corresponding paragraphs**
- **validation dataset: task2 labels (changes labels) and their corresponding paragraphs**

Reasons:

- **The model will be fine-tuned deeply because we have sufficient data to train.**
- **It can make the model tuning parameters in the direction of task2 when the validation dataset adopts the task2 labels. We should focus on task2 if we want to let three tasks can be achieved under a uniform framework because it can infer the task1 label and be applied to task3.**

Experimental setting

- **BERT-Base: 12-layer, 768-hidden, 12-heads, 110M parameters**
- **Max-length of input: 256**
- **Batch size: 32**
- **Loss: sparse categorical crossentropy**
- **Optimizer: Adam**
- **Learning rate: 2*10^(-5)**

Result

The result of validation set

Data set	Task1.F1	Task2.F1	Task3.F1
Validation set	0.85542	0.75193	0.39669

Task3 is a difficult task because the predicted Task3 labels inevitably have a chain reaction. Once there is a predicted error in the labels, it may lead to all the subsequent labels being wrong. That is error accumulation, which leads to a lower result in Task3.

e.g.	**task3 label:**	**1**	**2**	**2**	**3**
	task3-binary label:	**1**	**[1]**	**[1,0]**	**[1,1,1]**
	task3-binary predict label:	**1**	**[1]**	**[1,1]**	**[1,1,1]**
	task3 predict label:	**1**	**2**	**3**	**4**

Result

Finally, we obtained the F1 scores, which are 0.75, 0.75, 0.50 in task1, task2, task3, and ranked first in task2 and task3.

Results

Team	Task1.F1	Task2.F1	Task3.F1
Zhang et al.(Our)	0.753	0.751	0.501
Strom	0.795	0.707	0.424
Singh et al.	0.634	0.657	0.432
Deibel et al.	0.621	0.669	0.263
Nath	0.704	0.647	---

Thanks !